An Integrated Approach to Earth Science Observation Scheduling

Robert A. Morris, Jennifer Dungan, Jeremy Frank, Lina Khatib, David E. Smith
NASA Ames Research Center
Address: MS 269-1 Moffett Field CA 94035
Email: morris@email.arc.nasa.gov

Abstract- As the number of Earth observation missions grows, and the capabilities of Earth observing science instruments improve, the volume of requests for high quality science data will increase, and with it, the importance of effective management of the process of acquiring the data. This process includes selecting observations, storing images, and downlinking data to ground stations. This paper describes a system whose primary components are a central scheduler, which generates sequences of high priority, scientifically useful science observations for a fleet of earth observing satellites, and, in addition, an on-board schedule revision system for updating schedules based on predicted or observed changes in the utility of observations.

I. INTRODUCTION

The Committee on Earth Observation Satellites (CEOS) estimates that international space agencies are planning more than 80 earth observing missions over the next 15 years. These missions will carry over 200 different instruments, providing measurements of many environmental change parameters. The commercial sector is also planning to launch several systems in the next five years that could provide complementary data [1].

Current practice in mission management assigns a separate team of science and operations personnel for managing each specific mission, with very little coordination and communication across missions. The need for coordinated imaging demands a change in this way of managing missions, with more communication among mission planners.

This paper describes research into the potential use of automated scheduling technology in addressing issues in coordinated science activity, focusing both on effectively managing the increasing volume of data, and increasing the scientific utility of the data acquired.

II. OBSERVATION SCHEDULING

Science observation scheduling for Earth Observing Satellites solves the following core problem: given a set of candidate observation requests, each with a priority or utility corresponding to an expected value of acquiring the corresponding image, a set of times for acquiring the images, and a set of constraints associated with how the images can be acquired, generate a set of assignments of observations to times that satisfies the constraints and maximizes the sum of the utilities of the observations. Constraints include

- *Solid State Recorder (SSR) capacity*: an SSR can only store a limited amount of image data;
- *Pointing* requirements (for slewable instruments): adequate time must be allocated between observations to allow an instrument to be slewed to the angle required by the next observation; and
- *Instrument duty cycle*: a science instrument has limits on the length of time it can be turned on, to ensure its safety and longevity.

In addition, science campaigns in the future will often involve data acquired from multiple imagers. For example, [2], the scientific question *what is the role of clouds and aerosols in heating and cooling of the global climate?* will be answered by radiation budget measurements taken by CERES, as well as identification of regions of subvisible cirrus taken by MODIS IR, both members of the PM Constellation. Such requirements for the coordination of science observations suggest the need for a scheduling system that has complete knowledge of the capabilities and status of a collection of instruments and all data requests submitted for these instruments.

To be effective, a scheduling system should be able to revise a schedule at any time, to take account of changes in observation utility due to unforseen cloud cover, unexpected events such as floods or eruptions, new data requests, or unanticipated changes in satellite resources. Achieving these capabilities on a single ground-based system is not feasible. Typical Earth observing satellites cannot communicate directly with each other, and can only communicate with ground stations about 5% to 10% of the time. Because of the limited communication windows, a ground-based scheduler would have little or no opportunity to revise the schedule in response to the contingencies that may arise.

The principle argument for an additional *on-board scheduling capability* is that the desirability of making an observation can change dynamically and unexpectedly, because of changes in meteorological conditions (e.g. cloud cover), unforeseen events such as fires, floods, or volcanic eruptions, or changes in satellite or ground station capability. For example, if a desired visual scene is completely obscured by clouds, then there is little point in taking it. In this case, satellite resources, such as power and SSR capacity can be better utilized taking another image that is higher quality. Likewise, if an unexpected but important event occurs (such as a fire, flood, or volcanic eruption), there may be good reason to take images of it, instead of expending satellite resources on some of the lower priority scheduled observations. Finally, if there is unexpected loss of capability, it may be impossible to carry out the schedule of planned observations. For example, if a ground station goes down temporarily, a satellite may not be able to free up enough SSR capacity to continue with the remaining schedule of observations.

Therefore, we propose an architecture for a science observation scheduling system that consists of two main components:

- A centralized scheduler for multiple satellites, and
- An onboard schedule revision system for each individual satellite.

These components interact as follows. A set of complete sequences of observations generated by the central scheduler is uplinked to each satellite during its communication window, along with a set of alternative observations. Once uplinked, the on-board system will receive inputs consisting of either updated weather predictions, or data analysis results, which will allow it to revise the expected quality of the nominal schedule. This revision could result in a change in the sequence of acquired observations, the result of choosing from the set of alternatives. After the data is acquired, the images are downlinked, and the central scheduler is notified of any modifications made to the nominal schedule by the revision system. This provides part of the input to the next scheduling cycle.

The remainder of this paper describes each of the two main components of this approach in detail.

III. THE CENTRAL SCHEDULER

It is likely that science observation scheduling is too difficult to be solved in general by algorithmic techniques that guarantee optimality, such as branch and bound. Heuristic approaches to solving observation scheduling problems include local search (SPOT scheduler, [6]) or greedy constructive methods (Landsat 7 scheduler, [3]).

Greedy search requires a heuristic that orders the decisions made by the scheduler in the process of building a complete, consistent schedule. In one version of greedy search, the heuristic informs the scheduler as to what request should be added next to the schedule, and when the observation should be taken. A common heuristic for ordering requests is in terms of priority, with higher priority requests added to the schedule first. This forms the basis for many of the previous heuristics used in observation scheduling, for example, Spot scheduling [6].

Another heuristic we have considered for a central scheduler is based on *contention* for either observation start time, or space on the SSR. The idea behind this heuristic is that it is sometimes possible to schedule more observations, thereby achieving a higher quality schedule, if observations are ordered based on a combination of priority as well as how hard they are to schedule based on contention with other requests for the same time slots, or for memory on the SSR. See [7] for a detailed discussion of contention heuristics.

Recently, advances have been made to greedy search techniques by adding an element of randomness to the selection process. An example of this is used in the central scheduling process. Heuristic-biased Stochastic Sampling (HBSS) [5] is an approach to solving constraint satisfaction problems such as observation scheduling. Instead of always making decisions based on the heuristic advice, HBSS probabilistically biases each decision based on the heuristic rankings of the possible choices, thus allowing for the possibility of decisions that do not follow heuristic advice. The probability that the scheduler does follow the heuristic advice is a tunable parameter called the *bias;* the bias we used here is a function that maps observation priorities to the probability that the request will be selected next for scheduling. Utilization of the bias allows the scheduler to potentially compensate for limitations in the general usefulness of the heuristic, and to allow for sampling of the solution space. The belief is that this way of achieving a balance between exploration and exploitation of heuristic advice will yield improved schedules.

Based on these ideas, the central scheduling algorithm can be described as follows:

1. For a fixed number of samples do
2. Build Schedule with HBSS
3. Plan remaining supporting activities
4. Save best results

The call to HBSS (2) is the first phase of the central scheduling process; in the second phase (3) a planner is called to create and schedule support activities related to slewing the instrument and downlinking images. Of the two phases, the HBSS phase is more computationally intensive, so we will describe it in more detail. It is described by the following algorithm:

1. While requested observation list is not empty
2. Pick a request
3. While there are places to put the request
4. Pick a time to take the observation
5. If no constraints are violated
6. Assign the time to the request
7. If the request has been assigned
8. Check forward

This algorithm performs two varieties of constraint checking, backwards (step 5) and forwards (step 8). Backwards checking is so-named because it checks for constraint violations between the current assignment of observation to start time with previous assignments already made. Forward checking, by contrast, compares the current assignment with requests not yet selected. Start times that can no longer be assigned to unselected observations as the result of the current assignment are deleted from the list of possible times for those requests.

IV. A SCHEDULE REVISION SYSTEM

During execution, a nominal schedule produced by the central scheduler is to be preferred in the absence of any changes in the actual or expected values of observations. Consequently, the behavior of the schedule revision system will allow it to revert to the ground schedule in the absence of any changes in the utilities of observations. It is further assumed that the satellite receives updates on the actual value of observations just completed (e.g. as the result of performing a cloud cover analysis of the acquired data on-board), or updates on the expected values of observations that could be done in the near future (via communication with other satellites, forward looking instruments, or weather forecast update).

The approach to on-board schedule revision is for a system to acquire more observations than it expects to be able to keep (given on-board storage capacity limitations), and incrementally discard those of lesser value, as necessary, in order to retain observations of higher value. This over-commitment helps ensure that a full complement of useful observations will be collected, even if later scheduled observations turn out to be of low value. The bias towards acquiring observations in the original schedule is implemented by artificially raising the utility value of the pre-scheduled observations to guarantee they are higher than any extra observation. These observations are "removed" from the schedule and combined with the extra observations as inputs to an on-board scheduler.

The algorithm for schedule revision is the following:

Inputs

1. **a complete schedule produced by a ground-based scheduler and uplinked;**
2. **a set of additional observations that were not scheduled, and**
3. **the utility of each observation in the schedule and in the set of alternatives.**

Setup Procedure

1. **Artificially boost the utility values of scheduled observations by the maximum utility of the extra observations**
2. **Remove observations from the schedule and combine them with the extra observations as requests for the on-board scheduler.**

For each time slot *t*

1. **Consider the set *R* of requests that can be scheduled at time *t*.**
2. **Apply a lookahead strategy to assign a heuristic value to each request in *R*.**
3. **While there are still requests to consider in *R***
    - **a. choose *r* in *R* that has highest heuristic value**
    - **b. If SSR has sufficient capacity for *r*, acquire and assess the actual utility of *r***
    - **c. Else if SSR has insufficient capacity for *r***
        - **i. Let *W* be the set of past observations with lower utility than *r* and higher SSR allocation than needed**
        - **ii. If W is not empty**
        - **iii. Let *w* be a minimum utility in *W***
        - **iv. Discard *w* for SSR release**

v. Acquire and assess actual utility of *r*
vi. Else remove *r* from *R*

Starting with the first execution time of requests, at any time, we have past time requests (either satisfied or not), current request(s), and future requests. The algorithm decides what observations to schedule out of the set of current requests, aiming at highest overall schedule utility. The loop in the algorithm represents the progress of observation selection and execution while "next" time slot t is moving along the time horizon.

The algorithm is applied to a scheduling horizon starting at a given time slot t, and evaluates the set of requests that can be scheduled starting at t. At time t the SSR has a set of images stored from images acquired before t, and it is assumed that there are cloud cover analysis algorithms on-board that have been applied to the stored images in order to possibly update their utility.

Step 1 of the algorithm computes the set R of all requests at the next time slot t that do not conflict with past scheduled requests. Step 2 computes a heuristic value for each request in R in terms of the overall utility of the schedule that would result from executing it. The heuristic value of a request r is calculated based on the utilities of "some" future requests that can be executed given that r is executed.

We have considered two approaches to devising a lookahead policy, called *fixed* and *variable*. Briefly, a fixed strategy is applied to a request r with start time t, given a fixed lookahead depth d. A *heuristic value* of r is computed as the maximal sum of the utilities of requests than can be scheduled with r within the horizon determined by d. The idea of a variable lookahead strategy emerged to solve a problem with the fixed approach associated with a "horizon effect". In some cases it is possible, by greedily delaying an observation in favor of a later one with greater utility, that a more future-looking fixed lookahead makes worse decisions than a less future-looking one. A variable lookahead approach avoids this anomaly by making the lookahead horizon depend on the point of quiescence in the search for better schedules. If the lookahead for horizons of i, $i+1$, ... $i+j-1$ specify the same choice for next step, it is highly likely that this is the best choice. i and j are positive integers that represent, respectively, the lookahead horizon at which the quiescence starts and the length of the quiescence.

Thus, in the variable approach, h depends on two other factors: RQL (Required Quiescence Length) and MLH (Maximum Lookahead Horizon). The value assigned to h corresponds to the lookahead horizon such that the past RQL lookahead horizons have agreed on which request is the one with highest heuristic value. MLH is an upper limit on h, in order to control the absolute length of the lookahead. (See [8] for more discussion.)

For each request time t, the on-board scheduler uses one of these strategies to assign a heuristic value to each of the observations requested at t. Then Step 3 of the revision algorithm discussed earlier is performed for selecting observations. Step 3 applies the greedy strategy for schedule revision. If adding a request does not violate SSR capacity, it is simply added. If there is a violation, required storage space will be made available by discarding, and releasing the SSR storage space of, acquired images whose actual (analyzed) utility is less than the expected utility of the current request. If no such acquired images exist, then the next highest expected utility request in R is considered. The process repeats until either a request is added or there are no more requests left. Note that different requests may produce different amount of data and, as a result, require different amount of SSR storages. This complicates the process of selecting, among acquired images, the best "set" to discard. We attempt for one with minimum utility provided it has sufficient SSR release.

V. EXPERIMENTS

Tests have been conducted on both the central scheduler and the schedule revision system in order to determine the effectiveness of the ideas underlying the above approach to generating and executing high quality observation schedules. This section briefly discusses some of the results of those tests.

Currently, all tests have been conducted on problems generated by a system that simulates orbit characteristics for Earth observing satellites. A problem is generated from a specification of a number of parameters, including

- Number of candidate requests
- Length of scheduling horizon
- Number and capabilities of instruments
- Number of ground stations available for downlink.

These factors combine to determine the difficulty of the problem. In general, the most difficult problems (i.e., the ones requiring the most search to find high quality solutions) are those that combine a large number of requests, a large planning horizon and a small number of opportunities for downlink. In addition, slewable instruments tend to create more contention among candidate observations for viewing opportunities (i.e.

different scenes can be viewed at the same time at different angles).

Many of the problem instances generated were tested on a range of satellite models. A satellite model includes a specification of

- The slew capabilities of instruments, defined as slew rate + maximum viewing angle off-nadir.
- The storage capacity of the on-board Solid State Recorder (SSR), and
- The duty cycle constraints on science instruments.

The amount of satellite resources clearly also contribute to the difficulty of a given problem; in particular, a larger SSR means less contention for space among candidate requests, hence fewer potential conflicts. We have used the EUROPA planning system [3] for both constructing satellite models, and as the system for scheduling support activities after HBSS has made observation scheduling decisions.

A. Central Scheduler Experiments

The experiments with the central scheduler have focused both on the effectiveness of a sampling-based approach to augmenting greedy search using HBSS, and on a comparison between a contention-based heuristic for guiding greedy search and the more conventional priority heuristic.

For the purpose of realism, our starting point for testing has been problem instances with sizes comparable to those of the Landsat 7 daily scheduling problem [3], scaled up for handling multiple instruments and satellites. A typical problem instance has 500 requests, a scheduling horizon of 100K seconds, on 3 to 5 satellites, each with a single slewable instrument with a maximum pointing angle of 20 degrees off-nadir, and a satellite model with SSR capacity of between 50 and 200 images. For these tests, we have typically disabled the duty cycle constraint.

Runs conducted with problem instances having these characteristics have resulted in modest improvements in the quality of the schedules produced using sampling with HBSS (only around 1% for 20 samples is typical). That the results have not been more dramatic is indicative of the relative lack of contention for time slots in problems of this kind (even with large horizons, the average number of opportunities for a given candidate request is less than three, and a large number of candidates have only one opportunity; thus the chances for conflict are small). Consequently, there are very few optimal solutions, and good solutions tend to differ from one another only in a small number of assignments.

Similarly, a comparison between contention-based and standard priority-based heuristics indicates that ordering based on contention can consistently result in modest improvement in schedule quality. This improvement is also influenced by the number and distribution of opportunities for observations. For problem instances in which observation opportunities are few, and contention is not heavy, we have found that priority alone sometimes works best as a heuristic. The best approach might thus be one in which each sample of HBSS utilizes a different heuristic (referred to in the literature as a "portfolio" approach [9]). Future tests will investigate this option.

B. Schedule Revision System Experiments

In order to identify the usefulness of on-board rescheduling, we are studying the expected gain in the value of observations collected over those that would be taken if we just followed the schedule produced on the ground. This, of course, depends on the frequency and nature of the value revisions. So, more generally, we would like to know the net gain in the value of observations collected as a function of the frequency and nature of value revisions. For the particular algorithm proposed above we would also like to know how this value is affected as SSR capacity changes, the ground schedule bias is decreased, the size of the set of alternatives increases, or as lookahead changes.

We consider two different value revision scenarios. In the first, we suppose that there is on-board image analysis software, so that the actual value of an image is updated after an image is taken. With the above algorithm this primarily affects which observations will be discarded to make way for future observations. In the second scenario, we suppose that updates of expected value are received for observations to be taken in the future, as might occur if updated cloud cover forecasts were provided to the satellite. Note that this information impacts both which observations are taken and which ones are kept.

Our current experiments involve scheduling horizons of up to 9 hours and problem sizes of up to 300 requests. Before the proposed enhancements, such problem sizes could not be solved in a reasonable time. A reasonable variable lookahead strategy turned out to be RAL=3, which in general outperformed one with RAL=2, while either RAL=2 or RAL=3 outperformed no lookahead (RAL=0) by about 10%.

In general, a variable lookahead strategy seems to be faster than, and out perform, fixed lookahead. An intutive explanation for the speed improvement is that, in a variable lookahead, only a small lookahead horizon is needed at most steps and the maximum horizon is rarely reached. On the other hand, the reason that better results are obtained with variable lookahead is apparently the avoidance of the "horizon effect".

Figure 1 shows the percentage improvements, in terms of overall schedule utilities, when applying different types of lookaheads over applying no lookahead. Sample problems are randomly generated with horizons ranging from 900 to 30,000 seconds and number of requests ranging from 50 to 300. Negative entries show that applying lookahead could result in worse schedules than with no lookahead. Such cases appear to be rare. Note that the fixed lookahead with horizon 1 had 7.6% best improvement over no lookahead while other lookahead approaches were able to reach 14% improvement.

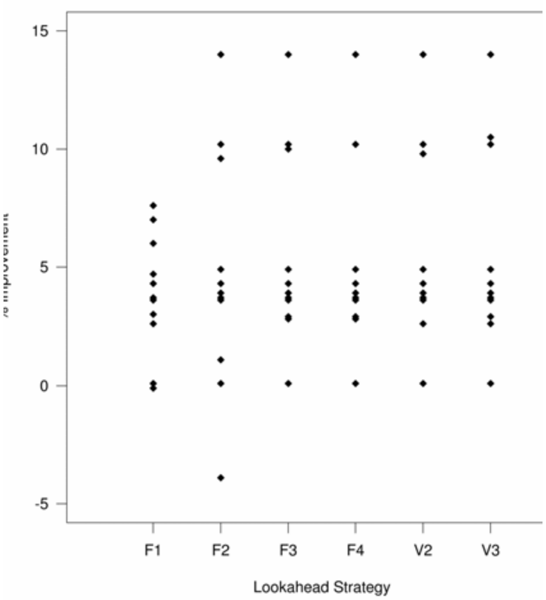

Figure 1. Percentage improvements of different lookahead strategies. The X-axis represent various lookahead strategies where Fi is fixed lookahead with horizon i=1, 2, 3, 4 and Vj is variable lookahead with acquiescence length j=2, 3. The Y-axis represent percentage improvement, when applying the corresponding lookahead strategy, over applying no lookahead, in terms of utilities of generated schedules.

VI. SUMMARY

This paper has proposed a new way of managing the selection of science observations for collections of Earth observing systems with different but overlapping science capabilities. This approach is motivated both by expected future requirements for coordinated observations, and by expected increasing demand for high quality science data. It has been argued here that communication delays between ground station and satellite, combined with an uncertainty in the conditions under which the data will be acquired, justifies an approach to scheduling that combines a robust central scheduler with limited on-board schedule revision.

References

[1] *Exploring our Home Planet: The Earth Science Enterprise Strategic Plan.* http://www.earth.nasa.gov/visions/index.html.

[2] M. Schoeberl. "The Afternoon Constellation: A Formation of Earth Observing Systems for the Atmosphere and Hydrosphere", *Proceedings of IGARSS 2002*.

[3] W. Potter and J. Gasch. "A Photo Album of Earth: Scheduling Daily Landsat 7 Activities". *Proceedings of SpaceOps'98,* Tokyo, Japan, 1998.

[4] D. Smith, J. Frank, and A. Jonsson. "Bridging the Gap Between Planning and Scheduling". *Knowledge Engineering Review, 15:1, 2000.*

[5] J. Bresina, "Heuristic-Biased Stochastic Sampling", *Proceedings of AAAI-96,* Portland, OR, 1996.

[6] J. Agnese, N. Bataille, E. Bensana, D. Blumstein, and G. Versaillie, "Exact and Approximate Methods for the Daily Management of an Earth Observation Satellite. In *Proceedings of the 5th ESA Workshop on Artificial Intelligence and Knowledge-based Systems for Space, 1995.*

[7] J. Dungan, J. Frank, A. Jonsson, R. Morris, D. Smith, "Planning and Scheduling for Fleets of Earth Observing Satellites", *Proceedings of ISAIRAS-2001*, Montreal, CA, 2001.

[8] L. Khatib, J. Frank, D. Smith, R. Morris, J. Dungan, "Interleaved Observation Execution and Re-Scheduling on Earth Observing Systems. *Proceedings of the ICAPS Workshop on Planning and Execution, 2003,* forthcoming.

[9] C. P Gomes, and B. Selman, "Algorithm Portfolio Design: Theory and Practice", *Proceedings of UAI-1997.*